```
non_nil_is_member  THM

⊢  ∀eq:{T≡₂}, L:T List. L ≠ nil ⇒ (∃x:T. x(∈eq) L)
| by Analyze 0 THEN Analyze 0 THEN Equivalence -1 THEN UnivCD
|
1. T : Type
2. eq : {T≡₂}
3. eq ∈ T→T→𝔹
4. ∀x:T. eq(x,x)
5. ∀x,y:T. eq(x,y) ⟺ eq(y,x)
6. ∀x,y,z:T. eq(x,y) ⇒ eq(y,z) ⇒ eq(x,z)
7. L : T List
8. L ≠ nil
⊢  ∃x:T. x(∈eq) L  by ListInd 7
|
|\
| 7. T List
| ⊢  nil ≠ nil ⇒ (∃x:T. x(∈eq) nil)
| | by Analyze 0 THEN Repeat (UnfoldTopAb -1)
| |
| 8. (nil = nil)→False
| ⊢  ∃x:T. x(∈eq) nil  by Assert (nil = nil)
| |
| 9. nil = nil
| ⊢  ∃x:T. x(∈eq) nil  by FwdThru -2 [-1] THEN Trivial
|
.\
. 8. u : T
. 9. v : T List
. 10. v ≠ nil ⇒ (∃x:T. x(∈eq) v)
. ⊢  u.v ≠ nil ⇒ (∃x:T. x(∈eq) (u.v))  by Analyze 0 THEN Witness u
. |
. 11. u.v ≠ nil
. ⊢  u(∈eq) (u.v)
. | by Witness4 u
. |    THEN
. |    FwdThru Thm* ∀P:𝔹. P ⟺ P = true₂    {assert_iff_btrue} [-1]
. |
. 4. ∀x,y:T. eq(x,y) ⟺ eq(y,x)
. 5. ∀x,y,z:T. eq(x,y) ⇒ eq(y,z) ⇒ eq(x,z)
. 6. T List
. 7. u : T
. 8. v : T List
. 9. v ≠ nil ⇒ (∃x:T. x(∈eq) v)
. 10. u.v ≠ nil
. 11. eq(u,u)
. 12. eq(u,u) = true₂
. ⊢  u(∈eq) (u.v)
. | by Rewrite (HigherC is_member_unrollC) 0 THEN HypSubst 12 0
. |    THEN
. |    Rewrite (HigherC ifthenelse_evalC THENC assert_evalC) 0
. |
. ⊢  True  by Trivial
END
```